CLAIMS

What is claimed is:

1. A method comprising:

setting a timer for a plurality of time intervals;

calling a polling function at the end of each of the plurality of time intervals, the polling function being performed by a first processor; and

if the polling function results in a positive result, processing the results of the polling function with a second processor.

2. The method of claim 1, wherein the polling function comprises polling a network stack.

3. The method of claim 1, wherein the first processor is an application processor

4. The method of claim 3, further comprising declaring the first processor to be dedicated to the polling function.

5. The method of claim 1, wherein the second processor is a bootstrap processor.

6. The method of claim 1, wherein a normal execution thread is processed by the second processor in parallel at least in part with performance of the polling function by the first processor.

7. The method of claim 1, wherein a timer interrupt is the only method of asynchronous event handling for the first processor and the second processor.

8. An event handling mechanism comprising:

a first processor, the first processor to perform a polling operation for event handling each time an interrupt timer reaches a specified time interval; and

a second processor, the second processor to perform a normal processing operation, the first processor to transfer data to the second processor for processing if the polling operation provides a positive result.

9. The event handling mechanism of claim 8, wherein the performance of the polling operation overlaps at least in part with the performance of the normal processing operation.

10. The event handling mechanism of claim 8, wherein the first processor is dedicated to event handling.

11. The event handling mechanism of claim 8, wherein an event for the polling operation comprises a network stack event.

12. The event handling mechanism of claim 8, wherein the first processor and the second processor are separate physical processors.

13. The event handling mechanism of claim 8, wherein the first processor and the second processor are logical processors in a single physical processor.

14. A computer system comprising:

a first processor, the first processor to perform an event handling function for the computer system;

a second processor, the second processor to perform a processing function for the computer system;

a timer, the timer being set for a time interval, a function call for the first processor being called at the end of the time interval; and

a memory, the first processor writing data relating to events to the memory to transfer the data to the second processor for processing.

15. The computer system of claim 14, wherein the second processor is a bootstrap processor.

16. The computer system of claim 14, wherein the first processor is an application processor.

17. The computer system of claim 14, wherein the first processor and the second processor operate in parallel at least in part.

18. The computer system of claim 14, wherein the timer provides the only event mechanism for the computer system

19. The computer system of claim 14, wherein the computer system comprises a single-threaded processing environment.

20. The computer system of claim 14, wherein the computer system is a multi-processor system.

21. The computer system of claim 14, wherein the computer system is a hyper-threaded system.

22. A machine-readable medium having stored thereon data representing sequences of instructions that, when executed by a processor, cause the processor to perform operations comprising:

setting a timer for a plurality of time intervals;

calling a polling function at the end of each of the plurality of time intervals, the polling function being directed to a first processor; and

if the polling function results in a positive result, directing the results of the polling function to a second processor.

23. The medium of claim 22, wherein the polling function comprises polling a computer interface.

24. The medium of claim 22, wherein the first processor is an application processor.

25. The medium of claim 22, wherein the instructions further comprise instructions that, when executed by a processor, cause the processor to perform operations comprising:

declaring the first processor to be dedicated to the polling function.

26. The medium of claim 22, wherein the second processor is a bootstrap processor

27. The medium of claim 22, wherein processing of a normal execution thread overlaps in time at least in part with performance of the polling function.